

## Change detection in streaming data analytics: A comparison of Bayesian online and martingale approaches

Bernadin Namono\*, Christos Emmanouilidis\*, Cristobal Ruiz-Carcel\*, and Andrew G Starr\*

\*Cranfield University, United Kingdom (email: {bernadin.namono; christosem; c.ruizcarcel; a.starr}@cranfield.ac.uk)

---

**Abstract:** On line change detection is a key activity in streaming analytics, which aims to determine whether the current observation in a time series marks a change point in some important characteristic of the data, given the sequence of data observed so far. It can be a challenging task when monitoring complex systems, which are generating streaming data of significant volume and velocity. While applicable to diverse problem domains, it is highly relevant to monitoring high value and critical engineering assets. This paper presents an empirical evaluation of two algorithmic approaches for streaming data change detection. These are a modified martingale and a Bayesian online detection algorithm. Results obtained with both synthetic and real world data sets are presented and relevant advantages and limitations are discussed.

**Keywords:** streaming analytics; change detection; martingale; Bayesian online detection;

---

### 1. INTRODUCTION

Engineering asset resilience management critically depends on the ability to detect events that may prevent assets from safely and dependably delivering their intended function. Complex asset monitoring produces data often characterised by significant volume and velocity. With the greater integration of internet of things technologies in such monitoring, data generation creates demands for considerable transmission bandwidth and significant computing resources. While the value of the data asset itself is increasingly acknowledged, collected data are often left poorly exploited, failing to take appropriate advantage of their potential for enhancing asset management performance (Kubler et al., 2015). Part of the challenge lies with the difficulty in understanding when any observable change in the data corresponds to events of interest, which in turn must require intervention actions. Directing the attention on significant events remains a challenging problem in asset monitoring. Terms such as outlier detection (André et al., 2008), novelty detection (Markou and Singh, 2003), and anomaly detection (Chandola et al., 2009), are all employed in this context and are relevant to change detection when monitoring engineering assets (Worden et al., 2000). However, collected data are typically not linked with validated event cases, making it hard to apply any supervised type of learning from data, making unsupervised (Filev et al., 2010) or semi-supervised types of learning more applicable in practice (Kingma et al., 2014). Most employed algorithmic approaches still require calibration and adjustment, as the dynamic characteristics of streaming data greatly vary across domains.

This paper performs an empirical evaluation of two change detection approaches, namely a non-parametric modified martingale type (Ho, 2005) and a Bayesian approach (Adams and MacKay, 2007). These are applied to event detection problems on both synthetic and real world data to enable better insight into their performance. The paper is structured as

follows. Section 2 places the present work in the context of the broader literature in the field. Section 3 presents a typical martingale change point detection algorithm, and introduces adaptations to address shortcomings of the original one. Section 4 describes the Bayesian online change detection algorithm. Results from the two approaches are presented and discussed in section 5. Section 6 is the conclusion.

### 2. RELATED WORK

Efficient change detection requires appropriate processing of streaming data to ensure that the delay time between a true change and its detection, as well as the rate of missed change events, are kept minimal (Ho, 2005). Depending on the availability of data annotated with labelled events, change detection methods can apply supervised, unsupervised, and semi-supervised learning and a range of optimisation methods.

In **supervised** learning-based methods, offline streaming data are labelled and then used to train models to perform change detection. These methods include classification or regression-based approaches, and it would be beyond the scope of this paper to mention all such applicable techniques. It is of interest though to highlight that major issues with this category of algorithms are their inability to detect unseen classes and their greediness in terms of data. Moreover, they require appropriate handling of unbalanced data, i.e. data with uneven numbers of pattern exemplars per class. For example, long time series with sparse events often present such learning challenges. In real world applications, the typical case is that there is a lack of annotated sample data which are representative enough of the range of possible circumstances and therefore a purely supervised approach is rarely applicable in practice.

**Unsupervised** learning based methods do not operate on labelled data and are therefore a more natural choice in practice. They typically seek to assign incoming data points into different clusters or simply to detect when an incoming

data point sufficiently deviates from a single or more clusters. A change is detected when consecutive data points are either assigned to different clusters or cannot be assigned to any cluster with a sufficient degree of confidence. The former may involve any type of clustering algorithms, whereas the latter typically rely on employing some type of distance metrics. Relevant methods include the cumulative sum (CUSUM) (Peach et al., 1995), martingale test (Vovk et al., 2003), minimax (Unnikrishnan et al., 2009), Bayesian inference based models (Adams and MacKay, 2007) (Ge et al., 2014) (Mohammad-Djafari and Féron, 2006), wavelets (Wang et al., 2018), as well as various forms of distance-based ones. Results from applying such methods are highly dependent on appropriately calibrating them to the problem at hand.

**Semi-supervised** learning, in which both supervised and unsupervised learning are combined for change point detection. Such methods are well suited for real world problems whereby sparsely labelled data are only available. Examples include feed-forward neural network (Zhang et al., 2017), used for offline training and the model built can be used online with non-negligible time delay. While they address the challenge of labelled data sparsity, they too rely on calibration. Change point detection problem can also be assimilated into an **optimisation** problem, wherein a cost function is minimised under some constraints (Truong et al., 2020). Many **heuristics** have also been developed to this end. Examples encompass sliding windows and bottom-up algorithm (SWAB) (Keogh et al., 2001), binSeg (Truong et al., 2020) in which global or local error cost is minimised. Appropriate choice of the cost function, sizing of the sliding window, and handling known challenges for the relevant optimisation approaches are among the typical challenges of such methods.

The choice of a change detection approach depends on the nature of the problem and the involved data characteristics. The motivation for the present work has been the need to develop streaming data change detection solutions for railway rolling stock asset monitoring. The aim is to evaluate current approaches (Namoano et al., 2019), study their shortcomings and strengths, and propose solutions and improvements to enhance change detection performance. Given this motivation, the paper focuses on fully online methods, i.e. methods that determine whether or not the current observation marks a change, given the data observed so far but not given the complete time series. The Bayesian online change detection (BOCD) (Adams and MacKay, 2007) and the martingale exchangeability test (Ho, 2005) are appropriate for such problems as they can handle univariate and multivariate data and can be adjusted to the problem at hand. For example, in BOCD, one can set the prior distribution of the underlying process or run a classical statistical test such as T-test to set hyperparameters. For the martingale methods, the computation of the so called p-values and the hypothesis test can be adapted to the problem. The two methods are presented next.

### 3. MARTINGALES AND EXCHANGEABILITY

#### 3.1. Basic Martingale algorithm

In a martingale process, the conditional expectation of the upcoming data point is the same as the current one, given all

observed values so far. The idea of the martingale concept of change detection is that by learning the statistical properties underlying the observed data, one can analyse any deviation by testing the data exchangeability. A time sequence of a vector of random variables  $\mathbf{X} = \{\mathbf{X}_i \mid i = 1..n \text{ and } n \in N\}$  is exchangeable if its joint distribution does not change by any alteration of the time sequence ordering of the observations  $\mathbf{X}_1 \dots \mathbf{X}_n$ . If the expected value of a sequence  $M$  is  $E(M_{n+1} \mid M_1..M_n) = M_n$ , where  $M = \{M_i \mid i = 1..n \text{ and } n \in N\}$  is a measurable function of  $\mathbf{X}$ , then  $M$  is a martingale with respect to  $\mathbf{X}$ . Considering a sequence of data  $\mathbf{S} = \{S_1, S_2, \dots, S_i, \dots, S_{n-1}\}$  where  $S_i$  is a multidimensional data point, a martingale test can be completed for each new point  $S_n$  involving three types of parameters as follows (Ho, 2005):

- 1) **Strangeness or non-conformity measure.** Denoted as  $\alpha_i$ , it expresses the dissimilarity between a specific point and other data points.
- 2) **p-values:** the p-value  $p_n$  for the new set  $\mathbf{S} \cup \{S_n\}$  defined as:  $p_n = \frac{\#\{i : \alpha_i > \alpha_n\} + \theta_n \#\{i : \alpha_i = \alpha_n\}}{n}$ , where  $\alpha_i$  is the strangeness,  $\theta_n$  is randomly chosen from  $[0, 1]$ ,  $i=1..n-1$  and  $\#\{\}$  is a counting function. A key property of the p-values is that as long a change does not occur, they are uniformly distributed in the interval  $[0, 1]$ .
- 3) **Computation of the martingale values**  $M_n^\varepsilon = \prod_{i=1}^n (\varepsilon p_i^{\varepsilon-1})$ , where  $\varepsilon$  is chosen from  $[0, 1]$  and controls the sensitivity of the change detection, while  $p_i$  are the p-values. The martingale values can be computed without storing all previous values as  $M_n^\varepsilon = \varepsilon p_n^{\varepsilon-1} M_{n-1}^\varepsilon$ . To remove the dependency on  $\varepsilon$ , a simple mixture of martingales can be used (Fedorova et al., 2012):

$$M_n = \int_0^1 M_n^\varepsilon d\varepsilon$$

When a change occurs, the p-values distribution becomes skewed, and the exchangeability condition is not met (Ho, 2005). The absence of change is represented by the hypothesis  $H_0: 0 < M_n < \lambda$  where  $\lambda$  is a defined threshold. The  $H_0$  is rejected when change occurs, hence  $M_n^\varepsilon \geq \lambda$ . Testing the exchangeability online involves computing power martingale values using p-values (Vovk et al., 2003). The main idea is to construct martingales which attain large values when small p-values are generated. The exchangeability test was applied to detect changes in sequential streaming data (Ho, 2005). This can work on unlabelled data streams, with multiple martingale tests with multiple strangeness and thresholds being used to determine whether a change occurs. Martingale difference values can be used to test whether a concept change occurred.

#### 3.2. Adaptations and modified martingale algorithm

Complex processes often exhibit such dynamic behaviour that selecting a constant threshold for the martingale test leads to higher false alarm rates. To reduce such effects, the martingale value can be reset at specified time intervals, when a change has not been detected (Balasubramanian et al., 2014). Furthermore, an adaptive threshold setting approach with Reproducing Kernel Hilbert Space (RKHS) projection (Wang et al., 2017) can be applied, but this does not estimate the value

of the initial threshold. Alternatively, different martingale tests with different thresholds and composite change detection criteria can be employed (Ho and Wechsler, 2007b). However, while this may improve performance, it also increases the risk of adding false alarms produced by individual martingale tests.

A modified martingale algorithm (Table 1) is introduced in this paper to address some of the challenges facing the original approach. The motivation for the modification is to address a common challenge for martingale approaches, related to the time delay between the true change point and the change point detected by the test. The potential significance of this often depends on the application context. In long run processes, the estimation of the strangeness becomes computationally expensive. This is due to the fact that when the frequency of changes is low or when there are no changes, the size of the buffer set of samples ( $T$  in Table 1), used to compute the strangeness, can become excessively large. To improve the computational efficiency of estimating the strangeness, down-sampling (Chawla, 2009) and windowing techniques can be used. Engineering processes often exhibit rapid changes in a limited number of time steps before going back to a steady-state. For example, a driver's acceleration is a typical case. A driving journey may switch between coasting and cruising to modes that involve gear change, acceleration and deceleration with braking. In such cases, the growth of the martingale value is not fast enough to capture the change. Lowering the detection threshold will result in confusing true changes with noise, resulting in higher false detections.

**Table 1 Modified Change detection basic algorithm used.**

**Algorithm 1** Online detection algorithm of change points

```

1: procedure
2:   Set the threshold  $\lambda_b$ 
3:   Set  $M_0 = 1$ , and  $G_0 = 0$ ,  $T = S = \{\}$ 
4:   Read the new feature  $x_i$ 
5:   Compute the strangeness  $s_i$  of  $x_i$  using  $T$ 
6:    $S = S \cup s_i$ 
7:   Compute the p-value  $p_i$  using  $S$ 
8:   Compute the martingale value  $M_i$  using  $p_i$ 
9:   Compute the  $G_i$  using  $M_i$ 
10:  Compute  $B_i = \max(G_i, M_i)$ 
11:  if  $B_i \geq \lambda_b$  then
12:    Mark  $x_i$  as change point
13:    Go to tep 3.
14:  else
15:     $T = T \cup x_i$ 
16:    Go to tep 4.
17:  end if
18: end procedure

```

Another difficulty is that the martingale value converges to zero for a streaming process with no changes. This often causes change points to be missed, as the growing struggle to reach the defined threshold. Moreover, when a change is detected, significant time delays may be observed. To reduce this effect, empirical tests carried out (Volkhonskiy et al., 2017) suggest that instead of basing the change determination on the original martingale  $M$  values, martingale growth values  $G$  can instead be utilised:

$$G_n = \max\{0, G_{n-1} + \log(M_n)\}$$

where  $M_n$  is the original martingale. Depending on the context, the martingale values  $G_n$  grows faster than the traditional martingale  $M_n$  and can hence, exhibit a reduced delay in detecting a change. The present paper introduces a further change in the way a change is detected via a decision variable  $B_n$ , computed as follows:

$$B_n = \max(G_n, M_n) \text{ with } B_0 = 0.$$

Hence, with this modified martingale the hypothesis  $H_0$  is rejected when  $B_n \geq \lambda_b$ , where  $\lambda_b$  a threshold, or accept it otherwise. This modification enables the algorithm to detect a change whether this is picked by growth in  $G_n$  or in  $M_n$ . Such modifications aim at addressing some of the aforementioned challenges of the original algorithm.

#### 4. BAYESIAN ONLINE CHANGE DETECTION

The BOCD approach applies online Bayesian reasoning by estimating for each current observation in a time series the probability to be a change point based on the data observed so far ( $X_{1:t}$ ). The underlying assumption is that the generated data are independent and identically distributed (i.i.d) random variables and the change point segments are not overlapping. These are not assumptions that can be assured to hold in practice. However, it allows a simplification in the involved estimations which in many cases can still offer adequate results. The underlying idea of the algorithm is computing the posterior probabilities  $P(r_t|X_{1:t})$  over the run lengths  $r_t$ .

$$P(r_t|X_{1:t}) = \sum_{r_{t-1}} P(r_t, r_{t-1}|X_{1:t})$$

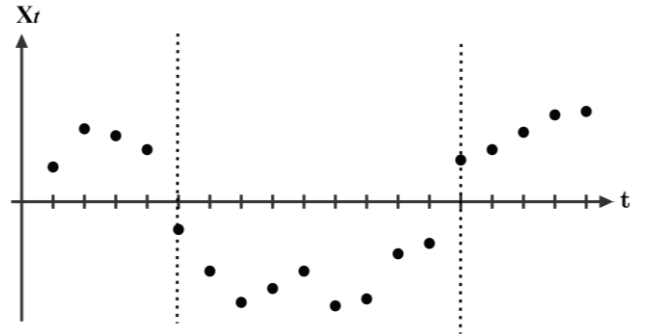


Fig. 1 Time series observations with two change point

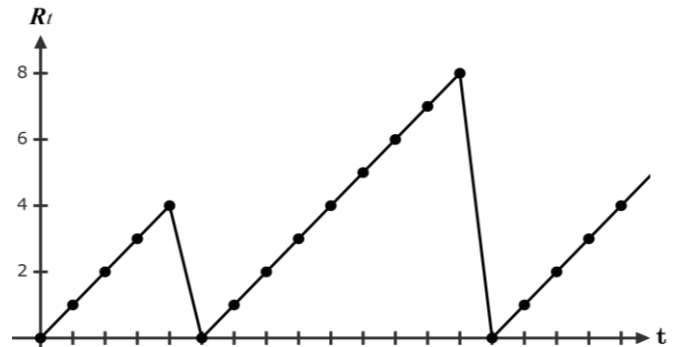


Fig. 2 Run lengths associated with the above two changes.

The run-length is increased by one when the present data point is determined to belong to the same distribution with the previous data. However, it is reset to zero when a change occurs, indicating that the present measurement point belongs to a new distribution (Fig. 1-2). The run length drops to zero when a change occurs. The corresponding estimated posteriors are shown in Fig. 3. Further details of the algorithm can be found here in the literature (Adams and MacKay, 2007).

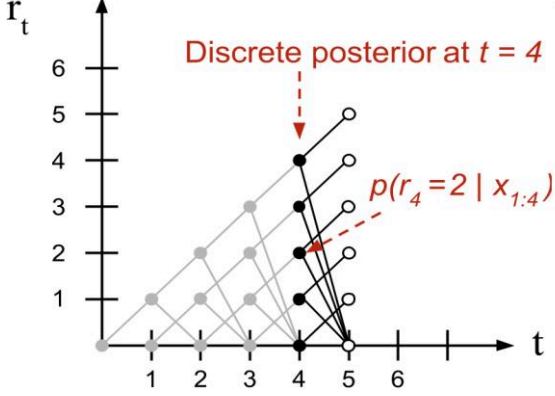


Fig. 3 Diagram of the message passing methods describing the posterior probabilities computed.

In practice, one of the drawbacks of B OCD methods is the quadratic growth of the run-length table with the growth of the size of the time-series. Pruning techniques exist to overcome this issue but may affect the efficiency of the algorithm.

## 5. EXPERIMENTS AND RESULTS

The experiments carried out aim to compare the modified martingale and the Bayesian online method and seek to identify performance characteristics for each approach. The work is motivated by industrial requirements to apply change detection on real world datasets from railway rolling stock monitoring. However, as the involved datasets are not publicly available, a choice was made to apply also the evaluated methods on publicly available data as well. For better evaluation of both methods, the present investigation used different data sources with different characteristics of change points (regular change points and random change points) and different dimensions and numbers of records. Specifically, two categories of datasets are used in the experiments. The first is a synthetic publicly available benchmarking dataset. Three types of synthetic data streams with induced changes are simulated in this dataset. The first is generated using normally distributed clusters data generator called NDC. It consists of a dataset with 1000 attributes, including 100,000 data points. The second is the USPS three-digit handwritten data, with 256 attributes and 7,291 data points. The third synthetic dataset is a modified version of nursery binary data (UCI) consisting of 5 attributes and 12,960 data points. For each dataset, change point occurs at every 1000 points. More details and the dataset are available online (Ho and Wechsler, 2007a). The second category of datasets represents real world datasets. The first dataset is one of the railway rolling stock monitoring datasets that motivated this work and is specifically from the train engines. It comprises three datasets representing the conditions of the engines hourly. Each dataset contains approximatively 3600 data points, taken every second. The attributes of the data

points include the engine frequency, the charge air pressure, the second stage oil, the exhaust gases pressure and temperature, and the ambient temperature and its pressure. The dataset contains 295 change points. Each change represents changing driving modes (idle, acceleration, maximum speed and deceleration). The second dataset represents the 2016 soccer UEFA championship eurogame with 16 games. The change points include game events, such as start, end, goals, and substitutions. The details on the data processing as well as its transformation are available on (Goutte et al., 2019).

### 5.1. Algorithmic parameters setting

For the modified martingale, an incremental SVM is used for the computation of strangeness, while  $\lambda_b$  was empirically set to 10, after experimentation. For the engines data as well as the eurogame datasets, the root mean square (RMS) method is used to compute the strangeness, while  $\lambda_b$  was empirically set to 14. In both cases, the k-means ( $k=2$ ) algorithm is employed to improve the change point location by separating the observations since the last change into two classes. The first class represents the statistical properties learned from the observations before the change and the second those after the change. For the martingale change detection, the algorithm of Table 1 is employed and when the adapted martingale value computed is greater or equal to the defined threshold, a change is detected. For the Bayesian online algorithm, the Gaussian distribution is used to update sufficient statistics referred to in (Adams and MacKay, 2007). Table 2 summarises the parameters used for both methods. The parameters were chosen via a nonlinear optimiser available in Matlab.

**Table 2 Parameters used to compute change points**

RMS: root means square,  $\lambda_b$ : martingale threshold  
hazard: The hazard function value used by the B OCD algorithm (Adams and MacKay, 2007).

Dataset	Martingale	B OCD
NDC	SVM, $\lambda_b = 10$	Gaussian, hazard=2000
USPS	SVM, $\lambda_b = 10$	Gaussian, hazard=1500
UCI	SVM, $\lambda_b = 10$	Gaussian, hazard=1500
Engines dataset	RMS, $\lambda_b = 14$	Gaussian, hazard=100
Eurogame	RMS, $\lambda_b = 14$	Gaussian, hazard=100

### 5.2. Evaluation approach

The standard performance assessment employed in typical classification problems is insufficient for change detection. In real world problems it can be highly misleading to detect a change too early or too late. Therefore, typical performance metrics are supplemented by an additional indicator that assesses the timelessness of the detection. Four performance indicators are therefore involved: recall, precision, the F1-score and the mean time delay (MTD). The precision gives the percentage of the correctness for a detection, meaning the likelihood of detecting true change. Recall measures the percentage of detected changes that are true changes. The F1-score is a metric of test accuracy. It measures the correctness of all identified cases. The MTD represents an average detection time lag, which is the duration between the true change and the one detected by the system.

$$Precision(P) = \frac{Number\ of\ Correct\ Detections}{Number\ of\ Detections} * 100$$

$$Recall(R) = \frac{Number\ of\ Correct\ Detections}{Number\ of\ true\ Changes} * 100$$

$$1 - score(F1) = 2 * \frac{P * R}{P + R}$$

### 5.3. Results and discussion

For each dataset, 50 runs have been performed, and then performance is averaged over all runs. These results are compared with the traditional martingale method (Ho, 2005). As seen in Table 3, the modified version of the martingale method has improved performance. The precision and the recall as well as the mean time delay have been improved in both synthetic and engines dataset.

**Table 3. Original vs modified martingale performance**  
\*\* and -- represent the modified and the original martingale methods respectively

Dataset	Precision	Recall	MTD (in s)
NDC data	**0.87 --0.85	**0.93 --0.90	**70 --92
Train engine data	**0.78 --0.60	**0.85 --0.80	**10 --50

**Table 4 Results overall employed datasets**

Datasets		Modified martingale				BOCD			
		P(%)	R(%)	F1(%)	MTD	P(%)	R(%)	F1(%)	MTD
Synthetic	NDC	92	98	95	40	55	82	65	2.85
	USPS	90	85	85	45	79	92	85	1.5
	UCI	85	92	92	80	79	92	85	90
Real world	Engines1	5	4	45	50	89	80	84	4.5
	Engines2	7	6	65	20	72	92	80	3.9
	Engines3	4	8	54	50	79	90	84	3.5
	CROESP	62	50	55	4.63	42	50	45	6.97
	ENGISL	29	12	17	3.5	50	35	41	3.91
	FRAALB	43	20	27	8.67	42	20	27	4.69
	FRAIRL	20	6	10	6	29	25	27	4.96
	FRAISL	36	25	30	5	47	50	48	8.38
	GERFRA	53	47	50	7.44	45	26	33	6.65
	GERITA	60	35	44	1	50	23	32	7.18
	GERPOL	14	8	10	2	47	54	50	6.2
	POLPOR	44	14	31	5	21	18	19	5.56
	PORAUT	38	15	21	4	45	25	32	4.97
	FRAPOR	50	25	33	6	21	15	17	3.11
	PORWAL	43	35	39	7	57	47	51	5.21
	RUSWAL	0	0	0	600	54	50	51	6.33
	SUIFRA	25	10	14	2	10	10	10	5.5
	WALBEL	50	39	44	5.57	41	39	40	9.27
	WALNIR	0	0	0	600	14	15	15	7.25

Table 4 summarises the results of the experiments. The colour represents a visualisation of the performance for each metric. Green indicates better performance. A straightforward observation is that in terms of time delay, the BOCD MTD is consistently better than the martingale. On the synthetic data with regularly spaced change points, the martingale method highlights better F1-scores in comparison to the BOCD. The precision shows that the BOCD is less robust in term of noise.

On the real world dataset experiments, as there is no regularity at the timing of the change points, it is observed that the

martingale method creates more false detections when fast paced changes occur in a short period (RUSWAL dataset result for example, in Table 4). Overall, BOCD outperformed the martingale, having lower mean time delay and better accuracy. Further work and more experiments are needed to replace the empirical choice for the calibration parameters with a systematic approach for their setting. Considering both types of datasets, the experiments show that when changes occur over long periods, the martingale is more accurate (F1-score) than the BOCD. However, for changes in short time windows, BOCD accuracy is higher than the martingale methods.

## 6. CONCLUSION

This paper proposes a modified version of the martingale method for change detection in multi dimensional streams as well as an empirical evaluation of both the modified martingale and Bayesian online change detection. The experiments show that the proposed modified martingale method is effective in real world as well as synthetic data. The comparison between the martingale method and BOCD shows that the martingale method achieved better performance regarding robustness to noise and detection accuracy when the observed time series is sparse in changes and the changes are of longer duration. BOCD is more accurate when changes of shorter duration occur, but generates high false alarm rates in the presence of noise. Future work will examine performance on real world datasets, using thresholds optimisation and mixing both methods through an ensemble learning process to enhance performance, taking into account the strangeness magnitude. Further research is needed to clarify the following:

**Strangeness measure:** For the employed data, unsuccessful attempts to use as strangeness measure distance-based metrics such as Euclidean and cosine have been made. It is therefore desirable to find a suitable or adaptive strangeness measure for the martingale methods so as to improve as well as the MTD.

**Modified martingale values:** Although empirical tests show better performance with the modified version, further research is needed for a sound justification regarding the conditions under which such improved performance is achieved.

**BOCD:** For this approach, there is a need for further research to improve its robustness and computational time complexity. This is needed to address the fact that when changes are sparse in time, the computational time complexity is quadratic.

## ACKNOWLEDGEMENT

The authors wish to thank Unipart Rail, Instrumentel and the Engineering and Physical Sciences Research Council (EPSRC) for their financial and technical support.

## REFERENCES

- Adams, R. P., and MacKay, D. J. C. (2007). Bayesian Online Changepoint Detection. Available at: <http://arxiv.org/abs/0710.3742>.
- André, N., Cho, H.-W., Baek, S. H., Jeong, M.-K., and Young, T. M. (2008). *Conformational analysis of a dinucleotide photodimer with the aid of the genetic*



- algorithm. Hoboken, NJ, USA: John Wiley & Sons, Inc. doi:10.1002/0471725382.
- Balasubramanian, V., Ho, S. S., and Vovk, V. (2014). *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications*. doi:10.1016/C2012-0-00234-7.
- Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey. *ACM Comput. Surv.* 41, 1–58. doi:10.1145/1541880.1541882.
- Chawla, N. V. (2009). “Data Mining for Imbalanced Datasets: An Overview,” in *Data Mining and Knowledge Discovery Handbook*, 875–886. doi:10.1007/978-0-387-09823-4\_45.
- Fedorova, V., Gammernan, A., Nourtdinov, I., and Vovk, V. (2012). Plug-in martingales for testing exchangeability on-line. in *Proceedings of the 29th International Conference on Machine Learning, ICML 2012*, 1639–1646. Available at: <http://alrw.net>.
- Filev, D. P., Chinnam, R. B., Tseng, F., and Baruah, P. (2010). An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics. *IEEE Trans. Ind. Informatics* 6, 767–779. doi:10.1109/TII.2010.2060732.
- Ge, D., Carraut, G., and Hernandez, A. I. (2014). Online Bayesian apnea-bradycardia detection using auto-regressive models. in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (IEEE)*, 4428–4432. doi:10.1109/ICASSP.2014.6854439.
- Goutte, C., Wang, Y., Liao, F., Zanussi, Z., Larkin, S., and Grinberg, Y. (2019). Eurogames16: Evaluating change detection in online conversation. in *LREC 2018 - 11th International Conference on Language Resources and Evaluation* (Miyazaki, Japan: European Language Resources Association (ELRA)), 1755–1761. Available at: <https://www.aclweb.org/anthology/L18-1277>.
- Ho, S. S. (2005). A martingale framework for concept change detection in time-varying data streams. *ICML 2005 - Proc. 22nd Int. Conf. Mach. Learn.*, 321–328. doi:10.1145/1102351.1102392.
- Ho, S. S., and Wechsler, H. (2007a). A Martingale Framework for Detecting Concept Changes in Time Varying Data Streams: Matlab Code and Data Sets. 1.
- Ho, S. S., and Wechsler, H. (2007b). Detecting changes in unlabeled data streams using martingale. in *IJCAI International Joint Conference on Artificial Intelligence* (San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.), 1912–1917. Available at: <http://dl.acm.org/citation.cfm?id=1625275.1625584>.
- Keogh, E., Chu, S., Hart, D., and Pazzani, M. (2001). An online algorithm for segmenting time series. *Proc. - IEEE Int. Conf. Data Mining, ICDM*, 289–296. doi:10.1109/icdm.2001.989531.
- Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. (2014). “Semi-supervised learning with deep generative models,” in *Advances in Neural Information Processing Systems*, eds. Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Curran Associates, Inc.), 3581–3589. Available at: <http://papers.nips.cc/paper/5352-semi-supervised-learning-with-deep-generative-models.pdf>.
- Kubler, S., Yoo, M. J., Cassagnes, C., Framling, K., Kiritsis, D., and Skilton, M. (2015). Opportunity to Leverage Information-as-an-Asset in the IoT-The Road Ahead. in *Proceedings - 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015*, 64–71. doi:10.1109/FiCloud.2015.63.
- Markou, M., and Singh, S. (2003). Novelty detection: A review - Part 1: Statistical approaches. *Signal Processing* 83, 2481–2497. doi:10.1016/j.sigpro.2003.07.018.
- Mohammad-Djafari, A., and Féron, O. (2006). Bayesian approach to change points detection in time series. *Int. J. Imaging Syst. Technol.* 16, 215–221. doi:10.1002/ima.20080.
- Namoano, B., Starr, A., Emmanouilidis, C., and Cristobal, R. C. (2019). Online change detection techniques in time series: An overview. in *2019 IEEE International Conference on Prognostics and Health Management, ICPHM 2019* doi:10.1109/ICPHM.2019.8819394.
- Peach, N., Basseville, M., and Nikiforov, I. V. (1995). *Detection of Abrupt Changes: Theory and Applications*. Prentice Hall, Inc. - <http://people.irisa.fr/Michele.Basseville/kniga/> doi:10.2307/2983416.
- Truong, C., Oudre, L., and Vayatis, N. (2020). Selective review of offline change point detection methods. *Signal Processing* 167, 107299. doi:10.1016/j.sigpro.2019.107299.
- Unnikrishnan, J., Veeravalli, V. V., and Meyn, S. (2009). Minimax Robust Quickest Change Detection. doi:10.1109/TIT.2011.2104993.
- Volkhonskiy, D., Nourtdinov, I., Gammernan, A., Vovk, V., and Burnaev, E. (2017). Inductive Conformal Martingales for Change-Point Detection. in *Proceedings of the 6th Workshop on Conformal and Probabilistic Prediction and Applications*, 1–22. Available at: <http://arxiv.org/abs/1706.03415>.
- Vovk, V., Nourtdinov, I., and Gammernan, A. (2003). Testing Exchangeability On-Line. *Proceedings, Twent. Int. Conf. Mach. Learn.* 2, 768–775.
- Wang, H., Zhou, F., Jiang, C., Qin, L., and Zhang, H. (2018). Change Point Detection for Piecewise Envelope Current Signal Based on Wavelet Transform. *J. Electr. Comput. Eng.* 2018, 1–10. doi:10.1155/2018/9529870.
- Wang, T., Lu, G.-L., Liu, J., and Yan, P. (2017). Adaptive Change Detection for Long-Term Machinery Monitoring Using Incremental Sliding-Window. *Chinese J. Mech. Eng.* 30, 1338–1346. doi:10.1007/s10033-017-0191-4.
- Worden, K., Manson, G., and Fieller, N. R. J. (2000). Damage detection using outlier analysis. *J. Sound Vib.* 229, 647–667. doi:10.1006/jsvi.1999.2514.
- Zhang, W., James, N. A., and Matteson, D. S. (2017). Pruning and Nonparametric Multiple Change Point Detection. in *2017 IEEE International Conference on Data Mining Workshops (ICDMW) (IEEE)*, 288–295. doi:10.1109/ICDMW.2017.44.

2020-12-18

# Change detection in streaming data analytics: a comparison of Bayesian online and martingale approaches

Namoano, Bernadin

Elsevier

---

Namoano B, Emmanouilidis C, Ruiz Carcel C, Starr A. (2020) Change detection in streaming data analytics: a comparison of Bayesian online and martingale approaches.

IFAC-PapersOnLine, Volume 53, Issue 3, 2020, pp. 336-341

<https://doi.org/10.1016/j.ifacol.2020.11.054>

*Downloaded from Cranfield Library Services E-Repository*